OpenCUDA+MPI

A Framework for Heterogeneous GP-GPU Cluster Computing

Kenny Ballou

July 20, 2013

1 Project Overview

2 Progress

1 Project Overview

- Terms and Definitions

2 Progress

- Node Power
- Node Configuration
- Sample/ Test Problem Development
- Results — N-Body Simulation

What is Distributed General Purpose Graphics Computing?

- Parallel:
  - Processing concurrently
- Distributed:
  - Processing over many computers
- GPU Computing
  - Highly Parallel Computing
- (Highly) Parallel + Distributed
  - Awesome
  - “High-Performance-Computing”

1 Project Overview
- Terms and Definitions

2 Progress
- Node Power
- Node Configuration
- Sample/ Test Problem Development
- Results — N-Body Simulation

Node Power

Requirements and Distribution

Now able to run all nodes

- 300 Watts per Node (Peak)
- ˜ 120 Volts at 20 Amps (Single Circuit)
- ˜ 48 Kilowatts

Salt Node Configuration

- Provisioning
  - Software
  - Configurations
  - Daemons/ Services
- Arbitrary Command Execution
  - Bring up and down nodes
  - Query load
- Complete for all nodes except for “master” or head node

Sample/ Test Programs

- 10^9 vector element-wise summation
- N-Body Simulation using particle particle, particle (adaptive) mesh (P3M) algorithm

Refactoring

```
def get_particles_in_grid(r, grid):
    def contains(r, d):
        return d[0] <= r[0] < d[0] + d[2] and \
               d[1] <= r[1] < d[1] + d[2]
    return (i for i in range(len(r[0]))
            if contains((r[0][i], r[1][i]), grid))
```

Refactoring — Result

```
def get_particles_in_grid(r, grid):
    x = np.intersect1d(
            np.where(grid[0] <= r[0])[0],
            np.where(r[0] < grid[0] + grid[2])[0],
            assume_unique=True)
    y = np.intersect1d(
            np.where(grid[1] <= r[1])[0],
            np.where(r[1] < grid[1] + grid[2])[0],
            assume_unique=True)
    return np.intersect1d(x, y, assume_unique=True)
```

Timing Results — N-Body Simulation

2k Times

Method	User (seconds)	Sys (seconds)	Real (seconds)
CPU	28.62	0.01	29.81
GPU	0.45	0.56	2.31
CUDA+MPI	N/A	N/A	N/A

Table : N-Body 2k Time Comparisons

Timing Results — N-Body Simulation

20k Times

Method	User (seconds)	Sys (seconds)	Real (seconds)
CPU	2368.39	1.30	2377.88
GPU	18.92	2.25	22.95
CUDA+MPI	1.19	1.01	2.94

Table : N-Body 20k Time Comparisons

Timing Results — N-Body Simulation

200k Times

Method	User (seconds)	Sys (seconds)	Real (seconds)
CPU	...	...	...
GPU	39.14	4.68	46.57
CUDA+MPI	6.43	5.01	13.65

Table : N-Body 200k Time Comparisons

Timing Results — N-Body Simulation

2m Times

Method	User (seconds)	Sys (seconds)	Real (seconds)
CPU	....	....	....
GPU	158.23	17.88	184.64
CUDA+MPI	68.50	44.93	127.04

Table : N-Body 2m Time Comparisons

Timing Results — N-Body Simulation

20m Times

Method	User (seconds)	Sys (seconds)	Real (seconds)
CPU	Nope	Nope	Nope
GPU	1159.89	147.24	1359.77
CUDA+MPI	623.41	156.82	901.62

Table : N-Body 20m Time Comparisons

OpenCUDA+MPI

A Framework for Heterogeneous GP-GPU Cluster Computing

Kenny Ballou

July 20, 2013